Towards a Standard Query Model for Sharing Decision-Support Applications

Walter Sujansky, Russ Altman, M.D., Ph.D.

Section on Medical Informatics, MSOB x215
Stanford University Medical School, Stanford, CA 94305
sujansky@camis.stanford.edu, altman@camis.stanford.edu

ABSTRACT

Many clinical decision-support applications are created in a centralized manner, but distributed widely for local use. When such applications include queries to electronic patient databases, the queries must be translated to conform to local database specifications. Because no well-defined standard model of clinical data exists, the translation process is ad hoc, costly, and error-prone. In this paper, we propose an abstract formalism, called the Standard Query Model Framework, for specifying a standard clinical data model and for supporting the automated and reliable translation of queries that appear in shared decision-support applications. We present the components of this framework, discuss their desirable features, and describe a prototype that we have developed for relational patient databases. We also highlight the outstanding research issues relevant to our approach.

INTRODUCTION

Although the medical informatics research community has devoted decades of work towards developing computerized decision-support tools, clinicians use relatively few clinical decision-support (CDS) applications today. Multiple technical and sociological reasons exist for this. Among the technical reasons are (1) the lack of integration of most CDS applications with existing clinical databases, which requires clinicians to re-enter into CDS applications many patient-specific data that are already recorded elsewhere, and (2) the diversity with which existing patient databases represent and retrieve data, which requires developers to customize or rewrite CDS applications that access databases if they wish to share these applications across provider sites. These technical obstacles create a trade-off with respect to deploying and using CDS applications: Applications that relieve users from re-entering patient data by automatically querying clinical databases must include low-level specifications regarding the implementation and the organization of those databases; these specifications vary significantly among existing clinical databases, so that the sharing of such applications across provider sites entails the extensive, complex, and error-prone translation of the database queries in the applications.

If significant heterogeneity among clinical databases persists, one can overcome this trade-off best by *automating* the translation of queries at specific database sites. Automation will allow CDS applications that access patient databases to be shared among provider sites without labor-intensive and error-prone manual customization. We envision a general framework for automating query translation, called the *Standard Query Model Framework*, that consists of the following components:

1. A standard reference schema of clinical data, with respect to which developers of CDS applications can formulate database queries in a site-independent way

2. A formal mapping language, which provider sites can use to represent the correspondence between their local database implementations and the site-independent reference schema

3. A translating compiler, which uses the mappings specified at each site to translate automatically the queries that appear in shared CDS applications to semantically equivalent queries that conform to the local database implementation.

In this paper, we propose a set of desiderata for the components of the standard query model framework, and we present an experimental prototype designed to fulfill the desiderata with respect to relational database implementations. Sections 2 and 3 describe how the heterogeneity of clinical databases currently inhibits the sharing of CDS applications and why past research to overcome database heterogeneity has not yielded an adequate solution. Section 4 presents the general components of the standard query model framework and discusses the desirable features of these components. Section 5 describes the design of TransFER, our prototype implementation, and Section 6 lists several research issues that must be addressed before the standard query model framework can be practically realized.

0195-4210/94/$5.00

QUERY MODEL HETEROGENEITY

The benefits of integrating CDS applications with clinical databases has been recognized for many years. In 1974, Shortliffe noted that integrating the MYCIN expert system with hospital information systems would allow more powerful rules to be added to the MYCIN knowledge base "without generating annoying questions for the physician" [1]. Recently, the Institute of Medicine's Committee on Clinical Practice Guidelines recommended the incorporation of practice guidelines into clinical information systems [2].

Despite these observations, there currently exist few CDS applications that are integrated with clinical databases. The heterogeneity of clinical databases makes it difficult for most institutions to integrate their local databases with CDS applications that have been developed elsewhere; the complexity of CDS applications makes it infeasible for most institutions to build their own applications, tailored to local database specifications. Medical informatics researchers can promote the wide-spread deployment of integrated CDS applications either by (1) enabling institutions to develop their own site-specific CDS applications, (2) standardizing every aspect of clinical databases, or (3) developing practical methods to "bridge" the heterogeneity of clinical database implementations. We believe that the latter approach will prove the most effective

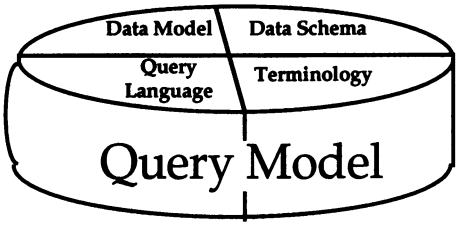

Figure 1. The components of a query model: The data model, the database schema, the query language, and the domain terminology

In the context of sharing CDS applications, bridging database heterogeneity entails resolving the differences between the *query model* that an application assumes and the *query model* that an operational clinical database provides. A query model [3] is the model of data representation and data retrieval that defines the interface between an application and a database. Specifically, a query model (see Figure 1) defines the abstract data model, the database schema, the query language, and the domain terminology that a database implements. Applications that interface to a database must specify queries in a manner that is consistent with the database's query model. Developers cannot currently write queries in CDS applications that will correctly retrieve data from arbitrary clinical databases because the query models of most clinical databases vary significantly [4]. Research to date has yielded no sound, economical, and generalized method to bridge query model heterogeneity so that CDS applications may be shared across provider sites.

PREVIOUS WORK

Researchers have pursued two general strategies for resolving heterogeneity between the query models of applications and databases: data translation and query translation. Data translation entails the transformation of *data* from the varied representations in which they may be collected to a common representation in which they can be accessed by CDS applications. Query translation entails the transformation of *queries* that appear in CDS applications into equivalent queries that are consistent with and can be executed directly against varied data representations.

Manual data translation, a technique used by certain clinical research databases [5], requires that medical records personnel manually abstract primary patient records into a common format that is subsequently uploaded into a centralized databank. The costs and delay of this method are generally prohibitive for routine decision-support. Automated data translation entails the algorithmic conversion of data directly to a common representation [6], or to an "interchange" format (such as HL7 [7]) that is subsequently translated to a common representation. Although automating data translation reduces the costs and delays of transforming data, the duplicate storage of data in the original and the common format increases the costs of operating an information system and compromises data integrity.

Query translation is preferable to data translation because no data need be duplicated and no delay is introduced before data is available to CDS applications. The Arden syntax, a standard and ostensibly portable knowledge-representation language for medical decision logic [4] requires *manual* query translation to share CDS applications. Under the Arden model, local database programmers translate manually each query that appears in a shared Arden program. The Arden syntax, however, includes few standard constructs for formulating database queries and no formal model of clinical data. In short, the Arden syntax specifies no standard query model. Because the representation of queries in the Arden model is, therefore, informal and because existing clinical databases are complex and highly variable, considerable effort may be required to

understand and to faithfully translate Arden programs when they are customized to local database environments [8] [9]. A codeveloper of the Arden syntax remarked upon this difficulty after a recent experiment in sharing Arden programs:

> "Although standards for representing clinical decision logic can be of great assistance in sharing the work of many, sharing may be delayed until common standards exist not only in the description of the logic, but in all aspects of the medical information system" [8].

This observation underscores the need to combine and expand current research in standardizing clinical data structures and clinical data terminology [10] to encompasses the development of a unified standard query model. A standard and well-defined query model is a necessary component of any technology for automating reliably the translation of database queries that appear in shared CDS applications. The automated and sound translation of queries will reduce significantly the costs, the delays, and the errors currently associated with sharing CDS applications. We define the *standard query model framework* as the research paradigm concerned with specifying a standard query model and with developing techniques to automatically translate queries based on this model.

THE STANDARD QUERY MODEL FRAMEWORK

The key component of the standard query model framework is a site-independent reference schema of clinical information that is specified using a **semantic data model**. The reference schema formally denotes the types of entities, the valid relationships among entities, and the terms used to denote entities in the domain of clinical medicine. Developers of CDS applications formulate all database queries with respect to this reference schema using a high-level query language. Provider sites use a **mapping language** to encode a set of mappings between the standard query model and their local query model. A **translating compiler** available at each site uses the mappings to translate the site-independent queries that appear in CDS applications to semantically equivalent local queries. The framework that we propose is graphically depicted in Figure 2. The advantage of this framework with respect to ad hoc methods is that it enables the systematic specification of mappings between a well-defined standard query model and a "target" database implementation; the translating compiler subsequently uses the mappings to translate an arbitrary number of queries automatically (eliminating the manual effort required to customize each query), and the translating compiler applies the same set of mappings to each query translation (ensuring that each query is translated consistently). We have identified design criteria for each component of the framework.

1. The **semantic data model** (SDM) [11] used to specify the reference schema must be sufficiently expressive, abstract, and well-defined. The SDM must represent all of the objects, properties of objects, and relationships among objects that typically appear in clinical information systems. For example, the standard relational model is not sufficiently expressive because it cannot explicitly represent associations among objects that are stored in separate relations. Also, the constructs of the SDM model must be sufficiently abstract to subsume the various ways in which data may be modeled in implemented databases. For example, the entity–relationship model [12] is not sufficiently abstract because it forces the reference schema to specify whether certain associations are modeled as relational attributes or as relational joins, a site-specific modeling decision. Lastly, the SDM must have formal semantics so that the meaning of the reference schema is unambiguous. To realize the benefits of a site-independent reference schema, the query language associated with the SDM must be equally expressive, abstract, and well-defined

2. The **mapping language** must be declarative. Declarative specifications facilitate the inspection, validation, and maintenance of encoded knowledge [13]. We believe that the specification and the management of mappings between query models will be complex and will require the assistance of computational tools. A declarative and formal language for the representation of mappings is a prerequisite for the development of such tools. Mappings specified as programs encoded in C or MUMPS are not amenable to automated inspection and validation.

3. The **translating compiler** must conserve the semantics of the mappings and must generate efficient queries. The compiler must apply the encoded mappings in such a way that if the individual mappings are correct, the query translation *in toto* will be correct. Because performance is an important consideration for applications that provide real-time decision support, sacrificing query efficiency in order to automate query translation is not a feasible trade-off.

THE TRANSFER METHODOLOGY

We have developed a prototype of the standard query model framework, called TransFER [3], that automatically translates queries to target

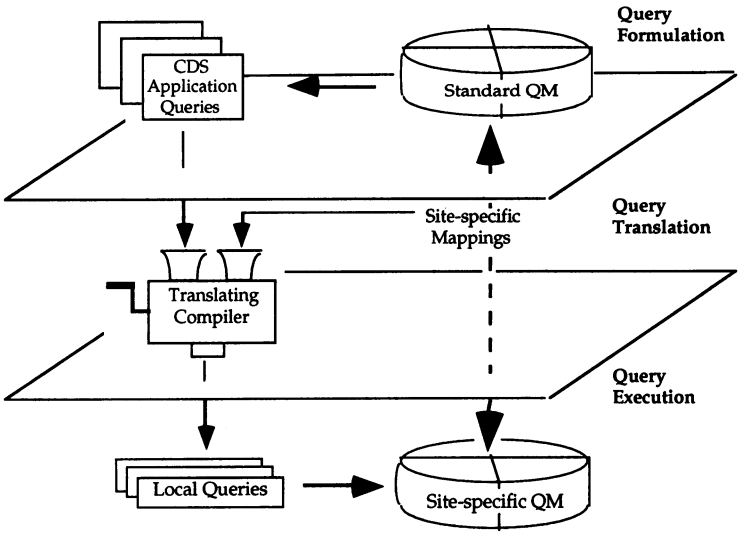

Figure 2. The standard query model framework. CDS application queries are formulated with respect to the standard query model ("Standard QM") using the high-level query language. No knowledge of any site-specific query model ("Site-specific QM") nor the query translation method is required at this level. A translating compiler performs the query translation based on encoded mappings that specify the correspondence between the standard query model and the relevant site-specific query model. The translated queries are semantically equivalent and can be executed by the local database.

heterogeneous relational databases. The TransFER methodology comprises four elements:

- A novel semantic data model, called FER (Functional Entity-Relationship model) for encoding site-independent clinical database schemas
- A query language, called ReFER, that corresponds to the FER data model and that allows users to specify data retrieval requests with respect to a FER schema
- A mapping language, called ERA (Extended Relational Algebra), that is based on the relational algebra [14] and that allows the constructs of an abstract FER schema to be mapped formally to equivalent constructs of a site-specific relational database schema
- A query-translation module, called TransFER, which applies ERA mappings to automatically translate ReFER queries into corresponding site-specific queries

Data Model and Query Language

The FER data model is designed to be sufficiently general to subsume diverse clinical database implementations. FER combines the semantic data modeling features of the entity–relationship (ER) data model [12] and the functional data model [15] to remedy the deficiencies of each model with respect to generality and expressiveness. The ER model distinguishes between attributes and relationships, so that ER schemas commit to a particular relational representation that, in fact, may vary among implemented databases. The functional model restricts the information that may be represented regarding associations among database objects; specifically, the model has no provisions for representing the attributes of functions, a capability that we have found useful for modeling the temporal semantics of legacy databases.

A sample clinical database schema encoded in the FER data model is illustrated in Figure 3. The schema illustrates the following modeling constructs of the FER data model:

Entity Types denote and describe *sets of objects* in the domain of discourse. For example, Patient and Name are entity types.

Entities denote *individual objects* in the domain of discourse. For example, Mr. Doe is an entity of the Patient and the Person entity types.

Relationship Types denote and describe *sets of associations* among members of entity types (that is, among entities). For example, the relationship type MD-Patient describes a set of associations between members of the Physician and the Patient entity types. Each binary relationship type defines a pair of directed functions, called Relationship Functions, that are used in ReFER queries to traverse the relationship type.

Relationships denote *instances of associations* among entities. For example, the tuple <Dr. Bob, Mr. Doe> denotes an instance of the MD-Patient relationship type.

IS-A Connections denote the generalization relationship between pairs of entity types. The semantics of IS-A connections imply set subsumption and relationship inheritance.

This minimal and abstract set of modeling constructs allows FER schemas to subsume query model

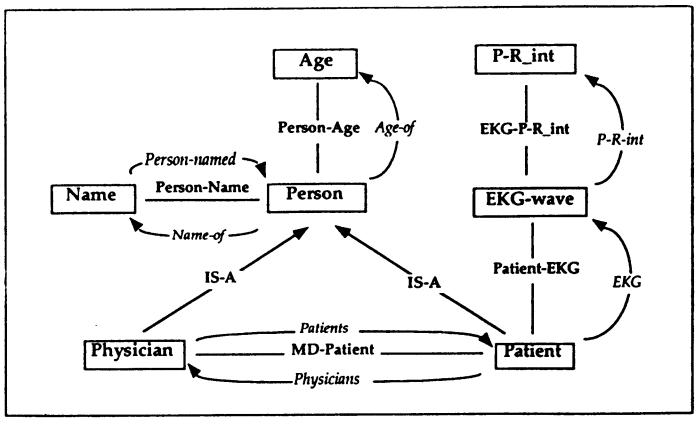

Figure 3. A sample FER schema. Boxes denote entity types; lines annotated with bold-faced text denote relationship types; arrows annotated with italicized text denote relationship functions; arrows annotated by IS-A labels denote IS-A connections. The schema conceptually represents a domain of discourse in which persons have names and ages, patients and physicians are subtypes of persons, patients may be associated with physicians, patients may have EKG test results, and EKG waves may have P-R intervals (clinically relevant features of EKGs). Two relational implementations of this conceptual schema appear in Figure 4.

heterogeneity among relational databases. For example, Figure 4 shows two different relational database schemas representing the domain of discourse encoded in Figure 3. Note that the schemas vary in several respects, including the identifiers used to denote certain entity types (`Doctor` versus `Physician`); whether data is stored or derived (`P-R-int` is explicitly stored in Schema 2 but must be derived from the value of `EKG` in Schema 1); and the representation of type hierarchies (the entity subtypes Patient and Physician are represented in different tables in Schema 1, but are included in the same table in Schema 2, distinguished by values of the `Type` attribute). Despite the representational heterogeneity of schemas 1 and 2, the FER schema in Figure 3 accurately denotes the *conceptual* contents of both schemas.

The syntax and semantics of the ReFER query language are based on the domain calculus [16] and derive their features from declarative query languages developed for the functional data model [15] and the ER data model [17]. The salient feature of the ReFER language is that it combines *declarative* and *functional* representations of query semantics, which allows queries to be expressed at an abstract level when the queries are formulated in CDS applications and later translated to the appropriate low-level operations when the queries are executed by specific clinical databases. For example, the following ReFER query retrieves the names of all patients with an EKG that has a P-R interval greater than 0.25:

name-of(pt) | (pt : Patient) AND
P-R-int(EKG(pt)) > 0.25

Note that the association between an EKG and its P-R interval is expressed as an invocation of the abstract function P-R-int() rather than as a join expression, a SQL select operation, or a foreign-function call. The representation of associations at this level of abstraction allows an association to be computed using whatever operations are indicated by the query models of existing clinical databases. The knowledge of which operations are required for specific clinical databases is represented locally using an extended relational algebra (ERA) mapping language.

Mapping Language and Query Translation

The ERA mapping language is based on the operators of the relational algebra: SELECTION, PROJECTION, CARTESIAN PRODUCT, UNION, and DIFFERENCE [14]. We have enhanced the basic relational algebra with syntactic and semantic constructs that increase its power to resolve representational heterogeneities among relational databases [3]. A mapping between the standard query model (for example, the FER schema in Figure 3) and a specific relational database implementation (Schema 1 or 2 in Figure 4) is defined by *assigning an ERA expression to each construct that appears in the FER schema.* The ERA expression represents the same semantics as the corresponding FER construct but is valid with respect to the relevant relational schema. For example, the entity type Patient and the relationship function Name-of from the FER schema of Figure 3 are assigned the following schema-specific ERA expressions.

Patient
(Schema 1) `Patient`
(Schema 2) SELECT [`Type = "PT"`] (`Person`)
Name-of (<arg>)
(Schema 1) PROJECT [`Name`] (<arg ERA>)
(Schema 2) PROJECT [`PName`] (<arg ERA>)

Note that the mappings for the FER construct Patient are distinct because schema 1 and schema 2 model the type hierarchy Person–Patient–Physician differently, and the mappings for the FER construct Name-of are distinct because the relational attribute "`Name`" appears in Schema 1 whereas the attribute "`PName`" appears in Schema 2. "<arg ERA>" denotes the ERA expression assigned to the *argument*

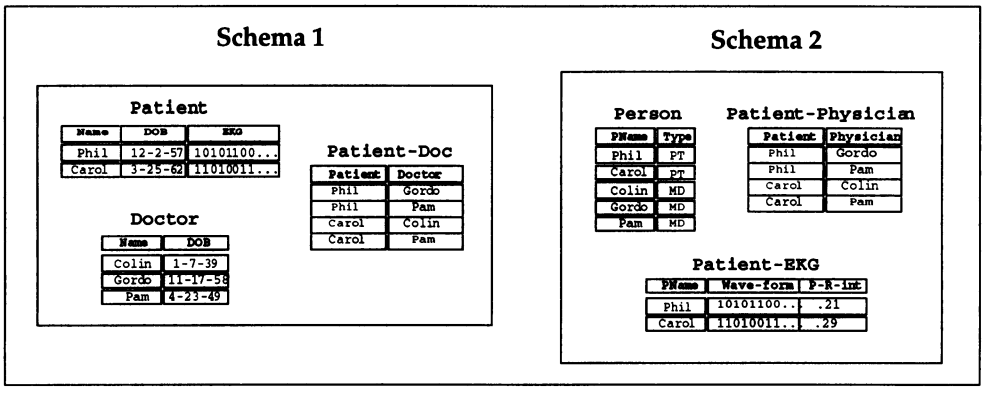

Figure 4. Two relational schemas representing heterogeneous implementations of the conceptual schema in Figure 3. Differences include the naming of entity types (Doctor versus Physician), the use of deroved versus stored attributes (P-R interval), and the representation of type hierarchies (Person–Patient–Physician).

of the Name-of relationship function in a specific query.

The translating compiler translates ReFER queries by composing the ERA mapping expressions that correspond to each construct appearing in the query (applying formal composition rules specified in [3]). For example, to translate the ReFER query

Name-of (pt) | (pt : Patient)

the compiler composes the ERA expressions corresponding to the Name-of construct and the Patient construct for the local schema. The result is a single schema-specific ERA expression that is semantically equivalent to the input ReFER query:

```
(Schema 1)  PROJECT [Name] (Patient)
(Schema 2)  PROJECT [PName] (
              SELECT [Type = "PT"] (Person))
```

The TransFER compiler subsequently modifies the expression, if necessary, to improve efficiency. The mathematically formalized semantics of relational algebra allow ERA expressions to be optimized in a sound and automated fashion. The compiler completes the translation by transcribing the resulting ERA expression to the dialect of SQL appropriate for the local relational DBMS (Oracle, Sybase, etc.).

We believe that the FER data model, the ReFER query language, the ERA mapping language, and the TransFER compiler fulfill the design criteria we have outlined for the standard query model framework. We currently are evaluating TransFER formally to test this hypothesis [18]. To meet the design criteria, we have constrained the TransFER methodology to accomodate relational databases only. This constraint allows us to take advantage of relational theory in defining the declarative semantics of the mapping language, in verifying the correctness of the translation process, and in optimizing the query expressions generated by the compiler. Although this constraint prevents clinical database sites that do not use relational technology from taking full advantage of the TransFER methodology, these sites still may benefit from the standard query model framework that we have defined: Non-relational sites will be able to customize CDS applications more consistently and more reliably (using existing methods) when the database queries that appear in CDS applications are encoded using the well-defined syntax and semantics of the standard FER schema and the ReFER query language. In any case, non-relational database sites will be no worse off than they are currently. At the same time, the significant and increasing number of clinical database sites that have relational interfaces will benefit substantially from the sound, efficient, and automated query translation that the TransFER methodology provides.

RESEARCH DIRECTIONS

Although the model we have developed is a sound foundation upon which to build a standard query model for sharing decision-support applications, at least three important research issues must be addressed before the standard query model framework can be practically useful: the incorporation of a standard medical terminology, the definition of sound temporal semantics, and the specification of a useful but finite reference schema of clinical data.

A prominent and difficult aspect of query model heterogeneity is the heterogeneity of medical terminologies. The framework we envision must include both a standard medical terminology as a seamless part of the standard query model and a powerful and sound method to automatically resolve differences between the standard terminology and the terminologies of legacy databases. Knowledge-based methods for representing and translating medical

terms [19] are most likely to yield abstract, declarative, and well-defined representations (that is, representations consistent with the desiderata of the standard query model framework).

Because the temporal aspects of clinical data and clinical queries are of paramount importance, the standard query model framework must include well-defined temporal constructs, as well as methods to map these constructs to the heterogeneous representations of temporal semantics in existing clinical databases. Das has defined a formal temporal semantics for relational clinical databases and a set of relational operators to resolve temporal heterogeneities among legacy databases [20]. We are currently investigating the application of these results to augment the TransFER methodology with temporal semantics and with the capability to resolve temporal heterogeneity.

The reference schema of clinical data must be sufficiently rich to support the data retrieval needs of many CDS applications, yet sufficiently general to subsume the data retrieval capabilities of most clinical databases. In defining such a schema, it may be useful to consider the design criteria enunciated by Gruber for the specification of shared domain ontologies [21]. Shared domain ontologies and reference database schemas are similar in content and in purpose: the conceptual representation of information to support the sharing of applications.

References

1. Shortliffe, E.H. *Mycin: A Rule-based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection.* Ph.D. thesis, Stanford University, 1974.
2. Field, M.J. and K.N. Lohr, editors. *Guidelines for Clinical Practice.* Washington, D.C.: National Academy Press, 1992.
3. Sujansky, W. *An Extended Relational Algebra for Bridging Representational Heterogeneity among Relational Databases.* Technical Report KSL-94-08, Section on Medical Informatics, Stanford University, 1994.
4. Hripcsak, G., et al., The Arden Syntax for Medical Logic Modules, in *Proceedings of the Symposium on Computer Applications in Medical Care,* R. Miller, Ed. Washington, D.C., 1990, pp. 200-204.
5. Weyl, Stephen, et. al., A Modular Self-describing Clinical Database System. *Computers in Biomedical Research.*, 8:279, 1975.
6. Marrs, K.A., et al. Unifying heterogeneous distributed clinical data in a relational database, in *Proceedings of the Symposium on Computer Applications in Medical Care,* C. Safran, Ed. New York, 1994, pp. 655-648.
7. HL7 Working Group. *Health Level 7 Interface Standard Version 2.1.* Philadelphia, 1989.
8. Pryor, T.A. and G. Hripcsak, Sharing MLMs: An experiment between Columbia-Presbyterian and LDS Hospital, in *Proceedings of the Symposium on Computer Applications in Medical Care,* C. Safran, Ed. New York, 1994, pp. 399-403.
9. Hripcsak, G., Desperately seeking data: Knowledge base-database links, in *Proceedings of the Symposium on Computer Applications in Medical Care,* C. Safran, Ed. *ibid.*, pp. 639-643.
10. Board of Directors of AMIA. Standards for medical identifiers, codes, and messages needed to create an efficient computer-stored medical record. *Journal of the American Medical Informatics Association.* 1:1, 1994.
11. Peckham, J. and F. Maryanski. Semantic Data Models. *ACM Computing Surveys.* 20(3): 153, 1988.
12. Chen, P.P.-S., The Entity-Relationship Model—Toward a Unified View of Data. *Transactions on Database Systems.* 1(1):9, 1976.
13. Genesereth, M.R. and N.J. Nilsson, *Logical Foundations of Artificial Intelligence.* Los Altos, CA: Morgan Kaufmann, 1987.
14. Codd, E.F., Relational completeness of data base sublanguages, in *Data Base Systems,* R. Rustin, Editor. New York: Prentice-Hall, 1972.
15. Shipman, D., The functional data model and the data language DAPLEX. *ACM Transactions on Database Systems.* 6(1): 140, 1981.
16. Pirotte, A., High Level Data Base Query Languages, in *Logic and Data Bases,* H. Gallaire and J. Minker, Editors. Plenum Press, 1978.
17. Hohenstein, U. and M. Gogolla. A calculus for an extended entity-relationship model, *in Proceedings of the Seventh International Conference on Entity-Relationship Approach..* Rome: Elsevier Science Publishing, Inc., 1989.
18. Sujansky, W and Altman, R.A., *Bridging the Representational Heterogeneity of Clinical Databases.* Technical Report KSL-94-07, Section on Medical Informatics, Stanford University, 1994.
19. Cimino, J.J. et. al., Knowledge-based approaches to the maintenance of a large controlled medical terminology. *Journal of the American Medical Informatics Association.* 1:35, 1994.
20. Das, A.K, et. al., A temporal-abstraction mediator for protocol-based decision support. Technical Report KSL-94-44, Section on Medical Informatics, Stanford University, 1994.
21. Gruber, T.R., *Toward Principles for the Design of Ontologies Used for Knowledge Sharing,* Technical Report KSL-93-04, Knowledge Systems Laboratory, Stanford University, 1993.